

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-318795

(43)Date of publication of application : 16.11.2001

(51)Int.Cl.

G06F 9/445

G06F 9/44

G06F 13/00

G06F 17/60

(21)Application number : 2001-052968

(71)Applicant : FUJITSU LTD

(22)Date of filing : 27.02.2001

(72)Inventor : OGURA KOUJIROU

(30)Priority

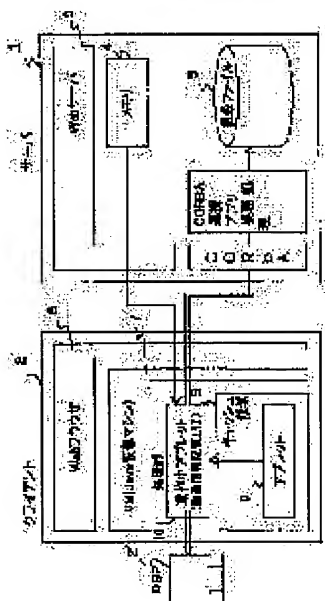
Priority number : 2000054056

Priority date : 29.02.2000

Priority country : JP

(54) SYSTEM AND METHOD FOR AUTOMATICALLY SETTING LATEST VERSION OF APPLLET

本発明は、最新のバージョンのアプリレットを自動的に設定するシステム及び方法に関する。



(57)Abstract:

PROBLEM TO BE SOLVED: To make it possible to always use an applet of the latest version by storing a program for version upgrade in a method on an applet side especially, and realizing the version upgrade of the applet by the processing of the applet itself in an automatic latest version setting device for automatically setting the latest version of the applet such as a Java applet in accordance with the request of a client.

SOLUTION: At the time of driving the Web browser 6 of the client 2 and executing an application through the Web server 3 of a server 1, the version is checked to see whether or not the applet used on the side of the client 2 is the latest applet and the applet is updated when it is not the latest applet.

Thus, by executing the processing every time the applet is

activated for instance, the Web browser is driven surely by the applet of the latest version at the time of using the applet, and by applying it to the Java applet for instance, the reliability of transactions is improved.

CLAIMS

[Claim(s)]

[Claim 1]A client and an applet latest version automatic setting system characterized by ***** characterized by comprising the following.

An applet memory measure which memorizes an applet of the latest version.

A server which has a version number memory measure which memorizes a version number of an applet memorized to this applet memory measure.

A version-number-information memory measure which processes according to an applet read from this server, and memorizes version number information of this applet.

A comparison means in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation,

An update means which reads an applet of the latest version from said applet memory measure, and updates an applet when this comparison result is not in agreement.

[Claim 2]A server which is provided with the following, reads version number information from said version number memory measure according to a read request of version number information, and is characterized by transmitting to a client.

An applet memory measure which updates an applet and always memorizes an applet of the latest version.

A version number memory measure which memorizes a version number of an applet memorized to this applet memory measure.

[Claim 3]A version-number-information memory measure which processes according to an applet read from a server, and memorizes version number information of this applet, A comparison means in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, A client having an update means which reads an applet of the latest version from said applet memory measure, and updates an applet when this comparison result is not in agreement.

[Claim 4]The applet latest version automatic setting system according to claim 1, wherein said applet is a Java applet.

[Claim 5]The server according to claim 2, wherein said applet is a Java applet.

[Claim 6]The client according to claim 3, wherein said applet is a Java applet.

[Claim 7]The applet latest version automatic setting system according to claim 4 being a time of said applet being in an active state at the time of said predetermined event generation.

[Claim 8]The client according to claim 6 being a time of said applet being in an active state at the time of said predetermined event generation.

[Claim 9]The applet latest version automatic setting system according to claim 1, wherein communication between said server and a client is performed by CORBA communication.

[Claim 10]The applet latest version automatic setting system according to claim 1, wherein said server is a bank operating center and said client is a terminal unit of a branch office.

[Claim 11]Claim 1, wherein said version-number-information memory measure is main memory or a disk, or an applet latest version automatic setting system given in four.

[Claim 12]Claim 3, wherein said version-number-information memory measure is main memory or a disk, or a client given in six.

[Claim 13]Applet memory processing which memorizes an applet of the latest version to an applet memory measure in a server, Version number memory processing which memorizes version number information corresponding to a version number of this applet to a version number memory measure in a server, Version-number-information memory processing which memorizes version number information read from said version number memory measure to a version-number-information memory measure in a client, Comparison processing in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, An applet latest version automatic setting method performing an update process which reads an applet of the latest version from said applet memory measure, and updates an applet when this comparison result is not in agreement.

[Claim 14]The applet latest version automatic setting method according to claim 13, wherein said applet is a Java applet.

[Claim 15]An applet memory storage function which memorizes an applet of the latest version to an applet memory measure in a server, A version number memory storage function which memorizes version number information corresponding to a version number of this applet to a version number memory measure in a server, A version-number-information memory storage function which memorizes version number information read from said version number memory measure to a version-number-information memory measure in a client, A comparison function in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, A recording medium storing a program which makes a computer perform an update function which reads an applet of the latest version from said applet memory measure, and updates an applet when this comparison result is not in agreement.

[Claim 16]Applet memory processing which memorizes an applet of the latest version to an applet memory measure in a server, Version number memory processing which memorizes version number information corresponding to a version number of this applet to a version number memory measure in a server, Version-number-information memory processing which memorizes version number information read from said version number memory measure to a version-number-information memory measure in a client, Comparison processing in comparison with version number information which read a version number of an applet of the latest version from said version number

memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, An applet latest version automatic setting program performing an update process which reads an applet of the latest version from said applet memory measure, and updates an applet when this comparison result is not in agreement.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention relates to the latest version automatic setting system of the applet which sets up the latest version of applets, such as a Java applet, automatically according to the demand of a client, and an automatic setting method.

[0002]

[Description of the Prior Art]The applet program driven on [other] soft is used today, and especially a Java applet is created in a Java language, is included in an HTML file, and is performed on a Web server.

[0003]Conventionally, in the client side, when it starts a Web page, the above-mentioned Java applet is downloaded from the server side, and on the Java virtual machine (Java virtual machine) of a web browser, the cache of it is carried out and it is used. After memorizing on the disks (hard disk etc.) of a client side, it may be sent to main memory and a Java applet may once be performed, although usually downloaded and used for the main memory of a client side.

[0004]

[Problem(s) to be Solved by the Invention]However, in the conventional system, after transmitting an applet to a client side, the applet by the side of a server may be rewritten by the latest version. Even in such a case, generally the renewal of an applet is not notified to a client side. For this reason, a client side will process using the applet of an old version.

[0005]In such a case, button grabbing of the renewal of the applet which a cache is already carried out and is driven can be carried out by the browser side, and re-reading of an applet can also be performed. However, there is no telling when there is any renewal of a version of an applet by the server side, but inefficient operation is needed. Where the cache of the applet is carried out depending on setting out of a browser, even if it performs re-reading, the applet written in, for example in the disk is only read, and upgrade of an applet may not be performed.

[0006]Then, this invention stores the program for upgrade in the method by the side of an applet in view of the above-mentioned actual condition, An applet's own processing realizes upgrade of an applet and the applet latest version automatic setting system which can use the applet of the latest version automatically, and an automatic setting method are provided.

[0007]

[Means for Solving the Problem]An applet memory measure which memorizes an applet of the latest version according to the invention according to claim 1 in an aforementioned problem, A server which has a version number memory measure which memorizes a version number of an applet

memorized to this applet memory measure, A version-number-information memory measure which processes according to an applet read from this server, and memorizes version number information of this applet, A comparison means in comparison with version number information which read version number information of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, When this comparison result is not in agreement, an applet of the latest version is read from said applet memory measure, and it can attain by providing an applet latest version automatic setting system comprising a client which has an update means which updates an applet.

[0008]Here, an applet memorized to the above-mentioned applet memory measure is the program created, for example with a Java language, it is updated one by one and an applet of the latest version is memorized. Version number information of an applet of the above-mentioned latest version is memorized by version-number-information memory measure.

[0009]Communication is performed to a server an applet of the above-mentioned latest version was remembered to be between web browsers which a Web server was set up, for example and were set as a client.

[0010]On the other hand, by an applet read from the above-mentioned server, a client side uses a web browser, for example, and performs animation processing and various display processing. A version number (version) of an applet in use is memorized by version-number-information memory measure by a client side. At the time of a predetermined event generation, a comparison means reads the newest version number information from a version number memory measure by the side of a server, and compares it with version number information memorized by version-number-information memory measure of a client side. And when the comparison result concerned is inharmonious, an update process of an applet of a client side will be performed (if it is a version number of a different applet).

[0011]By constituting in this way, an applet which can check an applet by the side of a server to predetermined timing, and is used by a client side can be automatically set as an applet of the latest version.

[0012]An applet memory measure which an aforementioned problem updates an applet according to the invention according to claim 2, and always memorizes an applet of the latest version, It has a version number memory measure which memorizes a version number of an applet memorized to this applet memory measure, version number information is read from said version number memory measure according to a read request of version number information, and it can attain by providing a server which transmits to a client.

[0013]Especially this claim charges a system configuration by the side of a server to a system given in above-mentioned claim 1. Therefore, composition of an applet memory measure in the above-mentioned composition and a version number memory measure is the same as the above.

[0014]A version-number-information memory measure which an aforementioned problem processes according to an applet which was read from a server according to the invention according to claim 3, and memorizes version number information of this applet, A comparison means in comparison with

version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, When this comparison result is not in agreement, an applet of the latest version is read from said applet memory measure, and it can attain by providing a client which has an update means which updates an applet.

[0015]Especially this claim charges a system configuration of a client side to a system given in above-mentioned claim 1. Said applet of a statement of claims 4 thru/or 6 is a Java applet.

[0016]By constituting in this way, a Web server can be set to the server side, a web browser can be set as a client side, and processing which uses a Java applet can be performed.

[0017]A statement of claims 7 and 8 is a time of said applet being in an active state at the time of said predetermined event generation. Here, cases where the above-mentioned applet becomes active are a case where an applet iconified, for example drives, and a case where a page which showed other Web pages is returned to a use page of an applet.

[0018]A statement of claim 9 is the composition that communication between said server and a client is performed by CORBA communication, in a statement of above-mentioned claim 1. In a statement of above-mentioned claim 1, an account server of beforehand of a statement of claim 10 is a bank operating center, and said client is a terminal unit of a branch office.

[0019]Banking is performed using an applet created by the operating center side, for example with a terminal unit of a branch office by constituting in this way, Since an applet of the latest version is read from an operating center to predetermined timing and an applet by the side of a window terminal can be updated when an applet is updated, an applet of the latest version can always perform business.

[0020]In a statement of claims 11 and 12, said version-number-information memory measure is main memory or a disk, for example. Here, main memory is RAM and a disk is a hard disk etc.

[0021]Applet memory processing which memorizes an applet of the latest version to an applet memory measure in a server according to the invention according to claim 13 in an aforementioned problem, Version number memory processing which memorizes version number information corresponding to a version number of this applet to a version number memory measure in a server, Version-number-information memory processing which memorizes version number information read from said version number memory measure to a version-number-information memory measure in a client, Comparison processing in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, When this comparison result is not in agreement, an applet of the latest version is read from said applet memory measure, and it can attain by providing an applet latest version automatic setting method which performs an update process which updates an applet.

[0022]This claim provides an automatic setting method of the applet latest version to a system given in above-mentioned claim 1. Also by constituting in this way, an applet which can check an applet by the side of a server to predetermined timing, and is used by a client side can be automatically set as an applet of the latest version.

[0023] Said applet of an invention of claim 14 is a Java applet. When this example explains an applet latest version automatic setting method, it is composition it is supposed that an applet to set up is a Java applet.

[0024] An applet memory storage function which memorizes an applet of the latest version to an applet memory measure in a server according to the invention according to claim 15 in an aforementioned problem, A version number memory storage function which memorizes version number information corresponding to a version number of this applet to a version number memory measure in a server, A version-number-information memory storage function which memorizes version number information read from said version number memory measure to a version-number-information memory measure in a client, A comparison function in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, When this comparison result is not in agreement, an applet of the latest version is read from said applet memory measure, and it can attain by providing a recording medium which stores a program which makes a computer perform an update function which updates an applet.

[0025] Applet memory processing which memorizes an applet of the latest version to an applet memory measure in a server according to the invention according to claim 16 in an aforementioned problem, Version number memory processing which memorizes version number information corresponding to a version number of this applet to a version number memory measure in a server, Version-number-information memory processing which memorizes version number information read from said version number memory measure to a version-number-information memory measure in a client, Comparison processing in comparison with version number information which read a version number of an applet of the latest version from said version number memory measure, and was memorized to said version-number-information memory measure at the time of a predetermined event generation, When this comparison result is not in agreement, an applet of the latest version is read from said applet memory measure, and it can attain by providing an applet latest version automatic setting program performing an update process which updates an applet.

[0026] Also by this claim's being the composition of specifying a recording medium which memorizes a program which performs automatic setting processing of the applet latest version to a system given in above-mentioned claim 1, and constituting in this way, An applet which can check an applet by the side of a server to predetermined timing, and is used by a client side can be automatically set as an applet of the latest version.

[0027]

[Embodiment of the Invention] Hereafter, an embodiment of the invention is described, referring to drawings. Drawing 1 is a system configuration figure of the applet latest version automatic setting system of this embodiment. In the figure, 1 is a server and 2 is a client connected to the server 1. Although two or more clients 2 are connected to the server 1, it represents and only a connection configuration with the client of 1 is shown. Here, Web server 3 is set to the server 1, and the version number file 5 which registers the memory 4 which registers the applet program (the number of the

latest versions) of the latest version, and the version number information of the corresponding newest applet is set to Web server 3. The applet registered into the above-mentioned memory 4 is a program for driving Java, and processes by driving the Java virtual machine later mentioned by the applet concerned.

[0028]The information on the version number of the corresponding latest version is registered into the version number file 5 with the gestalt shown, for example in drawing 2. Drawing 2 (a) is a figure showing the memory form of the version number file 5, it comprises 4 bytes, 2 bytes of low rank is version information, and top 2 bytes is level information. For example, in the case of the example shown in drawing 2 (b), it is "V20L10", and a version number shows "20" and a level shows the applet latest version of "10." Therefore, as shown in the figure (b) in this case, "14" is registered into the version area of the version number file 5, and "0A" is registered into level area. In the same version, the above-mentioned level does not need to show a development number, for example, and even a level does not necessarily need to include it in the latest version like this example in this invention.

[0029]On the other hand, in the client 2 side, the software of the web browser 6, Java virtual machine 7 (virtual storage system (VM)), and the applet 8 are set up, respectively, and the applet 8 is registered into the cache field 9. The version number information of a working applet is memorized by the treating part 10, for example, this version-number-information storage area is set to the predetermined address.

[0030]The web browser 6 carries out image display of the data of the HTML (hypertext makeup language) form supplied from Web server 3, and the applet (Java applet) 8 processes using Java virtual machine 7. The above-mentioned applet reads the applet of the latest version from the server 1 by the method mentioned later, and updates the applet written in the cache field 9. Display (DSP) 2' performs the display according to the drive of the above-mentioned applet.

[0031]Drawing 3 is the example which applied the relation between the above-mentioned server 1 and the client 2 to the business store server and window terminal of the bank. Here, the business store server 11 corresponds to the above-mentioned server 1, and the window terminal 12 corresponds to the client 2. The business store server 11 has Web server 3, the memory 4, and the version number file 5 as mentioned above, and performs CORBA (Common object request broker architecture) communication between the window terminals 12. For this reason, the cooperation application 13 for performing CORBA communication is also carried.

[0032]It is the composition which the above-mentioned web browser 6 is carried in the window terminal 12 side, and is started with the operator 14. When the applet used in this case is related with banking and an operator uses it at the bank window, it is composition provided with a required display and function.

[0033]In the applet latest version automatic setting system of the above composition, the processing operation is explained below. Drawing 4 is a flow chart explaining the processing operation of this example, and is a flow chart explaining the main process of an applet. In the following explanation, Web server 3 of the business store server 11 which is the server 1, and the web browser 6 of the window terminal 12 which is the client 2 assume that it has already started.

[0034]First, initial processing is performed (Step (shown by the following S) 1). This initial processing (init) is processing which starts the web browser 6, reads an applet from the server 1, and loads the applet of the latest version to the cache field 9 in the client 2 (window terminal 12). And the initial method specifically shown in drawing 5 is performed. That is, constant variable initialization processing is performed first (S2), and the version number (self-version number) of the applet loaded to the above-mentioned cache field 9 is set up. In this case, if the version of the present applet is "V20L10" shown in drawing 2, the data of "140A" will be written in, for example. Then, initial processing of an applet is performed (S3).

[0035]Next, version number check processing is performed (S4). Especially the version number check processing performed here is processing performed at the time of the start of an applet, and is performed in start method processing of an applet. Drawing 6 is a figure showing the processing, and version number check processing is performed within the above-mentioned start method (S4). Therefore, version number check processing (start) shows especially this processing.

[0036]Drawing 7 is a flow chart which explains the above-mentioned version number check processing concretely. First, version number reference is held to the server 1 (server 11 of a business store) (S5). And comparison processing with the version number information memorized by the memory of the server 1 is performed (S6), and when the version number of the applet read to the client 2 is older than the version number by the side of the server 1 (self-version number < server version number), end processing mentioned later is performed (S7 is Y (yes)).

[0037]Here, drawing 8 is a flow chart explaining the above-mentioned server version number inquiry processing and reading processing of a server version number. This processing is performed according to a server version number reference method. First, this processing initializes the CORBA (CORBA) communication (S5-1), and connects it to the client 2 (window terminal 12) side at the server 1 (server 11 of a business store) (S5-2). And the version number information of an applet is read from the version number file 5 in the server 1 (server 11 of a business store) (S5-3).

[0038]In this case, the version number file 5 in the server 1 (server 11 of a business store) is opened first (S5-4), and reading processing of a version number file is performed (S5-5). And the read version number information is written in (S5-6), and a version number file is closed (S5-7). Then, a server version number is set as an out parameter (S5-8). And connection with the server 1 (server 11 of a business store) is cut (S5-9), and the CORBA (CORBA) communication post-processing is performed (S5-10).

[0039]As mentioned above, although version number inquiry processing is performed, The above-mentioned version number check processing (start) is processing performed at the time of the start method processing after applet initial setting, the version number of the applet read to the client 3 is the same as the version number by the side of the server 1 (self-version number = server version number), and judgment (S7) is set to N (no). Therefore, in the version number check processing in a start method, it shifts to an event waiting state after that (S8).

[0040]Then, dealings are usually started. For example, the processing which passed on-line as opposed to the server 1 (business store server 11) from the client 2 (window terminal 12) side is charged. There is also processing which processing is performed by an offline state depending on a

work breakdown, and is performed only by the window terminal 12 side.

[0041]It is judged that waiting (S8 and S9 are N (no)) and an event generate generating of an event in the meantime whether it is a user event (S10). (S9 is Y (yes)) Here, if it is a user event, it will be judged whether on-line processing is required (S11). Here, a user event is processing which needs on-line processing as mentioned above, and is processing which connects with the server 1 (server 11 of a business store), and uses the application by the side of the server 1.

[0042]In this case, version number check processing (S12) is performed first. This version number check processing (S12) is performed according to the flow chart shown in above-mentioned drawing 7. That is, the version number reference (S5) to the server 1 (server 11 of a business store), the comparison processing (S6) of version number data, and also the version number of an applet to be used compare whether it is old as compared with the version number by the side of the server 1 (S7). (self-version number < server version number) Inquiry processing of a version number is performed according to the above-mentioned version number reference method (S5-1 of drawing 8 - S5-10).

[0043]The version number of the version number information read into the version number file and the applet read now is compared, event processing which a user will specify if it is the same version is performed (S13), and the above-mentioned result returns to an event waiting state (S9). In this case, the version number of the applet registered into the server 1 (server 11 of business store) side is not updated, It is the same version number as the applet by the side of the client 2 (window terminal 12), the applet used by the present client 2 side is also the newest applet, and the present applet performs terminal business, without updating an applet.

[0044]Drawing 9 is a figure showing the above-mentioned processing typically. In this case, since the applet of the latest version is memorized by the version number memory of the server 1 (server 11 of a business store) and the version number of the applet is not updated, version number information is the same as before. Therefore, after performing inquiry processing (** shown in drawing 9) of the above-mentioned version number information and judging coincidence of a version number, start processing of dealings is performed (** shown in drawing 9).

[0045]On the other hand, in the above-mentioned judgment (S7), when it has a relation of a self-version number < server version number (S7 is Y (yes)), it is judged as what the version number of the applet upgraded, and end processing is performed (S14). This processing is DESUTOROI processing (destroy) (S14), eliminates the program of the applet memorized to the cache field 9, and reads the applet of the latest version from the server 1 (S15).

[0046]The version number information of a version corresponding from the version number file 5 is read, and it writes in the memory by the side of the client 2. For example, in this case, it upgraded to the memory, for example, the data of "1500" or "140B" is written in.

[0047]In above-mentioned processing, even if an event generation occurs, when it is not on-line processing (for example, when performing processing which shifted to other Web pages which do not use an applet) (S11 is N (no)), corresponding user event processing is performed (S16).

[0048]On the other hand, for example, when the client 3 displayed other Web pages or the browser has iconified, An applet is in an inactive state, and when the browser iconified when it returned from this state to a Web page is restored, a system event occurs (S10 is N (no)). In this case, it is judged

whether it is the above-mentioned start event (S17).

[0049]Here, when starting a start method (S17 is YES (yes)), the above-mentioned version number check processing (start) is performed. It is performed by this processing according to the flow chart shown in drawing 7, and Version number inquiry processing (S5 and S5-1 - S5-10), Version number comparison processing is performed (S6, S7), when it is a self-version number < server version number, it is judged as what the version number of the applet upgraded, and end processing is performed (S7 is Y (yes) and S14). Therefore, the applet memorized to the cache field 9 in this case is eliminated, and the applet of the latest version is read and memorized from the server 1 (S15). The version number information of a version corresponding from the version number file 5 is read, and it writes in the memory by the side of the client 2.

[0050]Drawing 10 is a figure showing the above-mentioned processing typically, the applet by the side of the server 1 (server 11 of a business store) is updated in this case, and version number information differs. Therefore, after performing inquiry processing (** shown in drawing 10) of the above-mentioned version number information and judging the disagreement of a version number, the update process of an applet is performed and version number information is also rewritten (** shown in drawing 10).

[0051]By processing as mentioned above, the applet of the latest version is always memorized by the memory by the side of the client 2 (window terminal 12), and the applet of the latest version can perform image display and animation display.

[0052]When the above-mentioned system event occurs, if it is not a start event, it will judge whether it is a DESUTOROI (destroy) event (S19), and end processing will be performed if it is a DESUTOROI (destroy) event (S20). This end processing is processing which eliminates the program of the applet memorized by main memory as mentioned above, reads the applet of the latest version from the server 1, and is memorized.

[0053]In the above-mentioned judgment (S19), if it is not a DESUTOROI (destroy) event, directed event processing will be performed (S21). As mentioned above, when an applet is made into an active state according to this invention, Or the version number check of an applet is performed on the predetermined conditions at the time of the user event which needs on-line processing occurring, etc., and when it is not an applet of the latest version, it is certainly updated by the applet of the latest version. Therefore, the applet of the latest version can always drive on the web browser 6.

[0054]In this example, although version number check processing was considered as the composition performed in the above-mentioned start method etc., it is not necessarily limited to explanation of this example. The version information of the version number used by this example was constituted from version information and level information, as shown in above-mentioned drawing 2, but it may consist of only version information.

[0055]In this embodiment, although the server 1 of this system was made into the server 11 of the business store of a bank and the client 2 was constituted as the window terminal 12 of a bank, this invention can also be applied to other fields, such as insurance and a security, without being limited to the above-mentioned banking system.

[0056]The applet latest version automatic setting system of this invention may be realized using the

system shown in drawing 11. That is, as shown in the figure, CPU(computer) 15 performs an above-mentioned system by the program (data) supplied from internal RAM and the hard disk 16.

[0057]As shown in the figure, only not only in the program (data) supplied from internal RAM and the hard disk 16, It is good also as composition which processes according to the program (data) which delivers and receives data between the storages connected outside, for example, is supplied from the storages 17, such as a magnetic disk, magnetic tape, a floppy (registered trademark) disk, and an optical disc.

[0058]As shown in the figure, it is good also as composition which uses the program (data) sent by the donor 18 through a communication line.

[0059]

[Effect of the Invention]As mentioned above, as explained in detail, according to this invention, when using an applet, a web browser can be driven using the applet of the latest version.

[0060]It cooperates between the client and server like the on-line system of a bank, and when the operating dealings with important reliability are realized by a program with portability, such as a Java applet, the improvement in reliability of dealings has a big effect.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a system configuration figure of the applet latest version automatic setting device of this embodiment.

[Drawing 2](a) is a figure explaining the composition of the memory area which registers the version number information of the latest version. (b) is a figure explaining the example which wrote actual data in the memory area shown in (a).

[Drawing 3]It is a system configuration figure explaining the example which applied the system of this embodiment to the business store server and the window terminal.

[Drawing 4]It is a flow chart explaining the processing operation of this embodiment.

[Drawing 5]It is a flow chart explaining the processing operation of an initial method.

[Drawing 6]It is a flow chart explaining the processing operation of a start method.

[Drawing 7]It is a flow chart explaining version number check processing.

[Drawing 8]It is a flow chart explaining a server version number reference method.

[Drawing 9]It is a figure explaining a right-hand side link mechanism.

[Drawing 10]It is a perspective view of a link mechanism.

[Drawing 11]It is a figure explaining the example which uses a recording medium.

[Description of Notations]

1 Server

2 Client

3 Web server

4 Version number file

5 Version-number-information memory

6 Web browser
7 Java virtual machine
8 (It is working) Applet
9 Cache field
11 Display (DSP)
11 Business store server
12 Window terminal
13 Cooperation application
14 Operator
15 CPU (computer)
16 RAM, a hard disk
17 Storage
18 Donor

*** NOTICES ***

JPO and INPIT are not responsible for any
damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

P1

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-318795

(P2001-318795A)

(43) 公開日 平成13年11月16日 (2001. 11. 16)

| (51) Int.Cl. ⁷ | 識別記号 | F I | テマコード* (参考) |
|---------------------------|-------|--------------|-------------|
| G 0 6 F 9/445 | | G 0 6 F 9/44 | 5 3 0 P |
| 9/44 | 5 3 0 | 13/00 | 5 3 0 B |
| 13/00 | 5 3 0 | 17/60 | 2 3 6 E |
| 17/60 | 2 3 6 | | 2 3 8 |
| | 2 3 8 | | 2 4 0 |

審査請求 未請求 請求項の数16 O L (全 11 頁) 最終頁に続く

(21) 出願番号 特願2001-52968(P2001-52968)

(22) 出願日 平成13年2月27日 (2001. 2. 27)

(31) 優先権主張番号 特願2000-54056(P2000-54056)

(32) 優先日 平成12年2月29日 (2000. 2. 29)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 小倉 康二郎

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 100074099

弁理士 大菅 義之 (外1名)

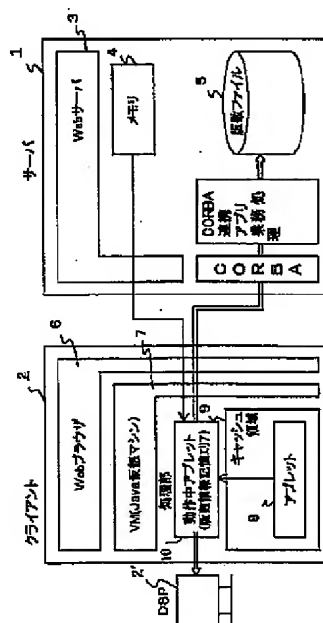
(54) 【発明の名称】 アプレットの最新バージョン自動設定システム、及び自動設定方法

(57) 【要約】

【課題】 本発明は、J a v a アプレット等のアプレットの最新バージョンをクライアントの要求に従って自動設定するアプレットの最新バージョン自動設定装置に関し、特にアプレット側のメソッドにバージョンアップ用のプログラムを格納し、アプレット自身の処理によってアプレットのバージョンアップを実現し、常に最新バージョンのアプレットを使用することが可能なアプレット最新バージョン自動設定装置を提供するものである。

【解決手段】 クライアント2のWebブラウザ6を駆動し、サーバ1のWebサーバ3を介してアプリケーションを実行する際、クライアント2側で使用するアプレットが最新のアプレットであるか版数チェックを行い、最新アプレットでなければアプレットを更新する。したがって、上記処理を例えばアプレットがアクティブになる毎に実行することによって、アプレットを使用する際必ず最新バージョンのアプレットでWebブラウザを駆動することができ、例えばJ a v a アプレットに適用することによって取引の信頼性を向上することができる。

本実施形態のアプレット最新バージョン自動設定装置のシステム構成図



【特許請求の範囲】

【請求項1】 最新バージョンのアプレットを記憶するアプレット記憶手段と、該アプレット記憶手段に記憶したアプレットの版数を記憶する版数記憶手段とを有するサーバと、

該サーバから読み出したアプレットに従って処理を行い、該アプレットの版数情報を記憶する版数情報記憶手段と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較手段と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新手段とを有するクライアントと、から成ることを特徴とするアプレット最新バージョン自動設定システム。

【請求項2】 アプレットを更新し、常時最新バージョンのアプレットを記憶するアプレット記憶手段と、該アプレット記憶手段に記憶したアプレットの版数を記憶する版数記憶手段とを有し、版数情報の読み出し要求に応じ、前記版数記憶手段から版数情報を読み出し、クライアントに送信することを特徴とするサーバ。

【請求項3】 サーバから読み出したアプレットに従って処理を行い、該アプレットの版数情報を記憶する版数情報記憶手段と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較手段と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新手段、とを有することを特徴とするクライアント。

【請求項4】 前記アプレットはJavaアプレットであることを特徴とする請求項1記載のアプレット最新バージョン自動設定システム。

【請求項5】 前記アプレットはJavaアプレットであることを特徴とする請求項2記載のサーバ。

【請求項6】 前記アプレットはJavaアプレットであることを特徴とする請求項3記載のクライアント。

【請求項7】 前記所定イベント発生時は、前記アプレットがアクティブ状態になる時であることを特徴とする請求項4記載のアプレット最新バージョン自動設定システム。

【請求項8】 前記所定イベント発生時は、前記アプレットがアクティブ状態になる時であることを特徴とする請求項6記載のクライアント。

【請求項9】 前記サーバとクライアント間の通信は、CORBA通信で行われることを特徴とする請求項1記載のアプレット最新バージョン自動設定システム。

【請求項10】 前記サーバは銀行営業センターであ

り、前記クライアントは支店の端末装置であることを特徴とする請求項1記載のアプレット最新バージョン自動設定システム。

【請求項11】 前記版数情報記憶手段は、メインメモリ又はディスクであることを特徴とする請求項1、又は4記載のアプレット最新バージョン自動設定システム。

【請求項12】 前記版数情報記憶手段は、メインメモリ又はディスクであることを特徴とする請求項3、又は6記載のクライアント。

10 【請求項13】 サーバ内のアプレット記憶手段に対して最新バージョンのアプレットを記憶するアプレット記憶処理と、

該アプレットの版数に対応した版数情報を、サーバ内の版数記憶手段に記憶する版数記憶処理と、

クライアント内の版数情報記憶手段に対し、前記版数記憶手段から読み出した版数情報を記憶する版数情報記憶処理と、

20 所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較処理と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新処理と、

を行うことを特徴とするアプレット最新バージョン自動設定方法。

【請求項14】 前記アプレットはJavaアプレットであることを特徴とする請求項13記載のアプレット最新バージョン自動設定方法。

30 【請求項15】 サーバ内のアプレット記憶手段に対して最新バージョンのアプレットを記憶するアプレット記憶機能と、

該アプレットの版数に対応した版数情報を、サーバ内の版数記憶手段に記憶する版数記憶機能と、

クライアント内の版数情報記憶手段に対し、前記版数記憶手段から読み出した版数情報を記憶する版数情報記憶機能と、

40 所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較機能と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新機能と、

をコンピュータに実行させるプログラムを格納することを特徴とする記録媒体。

【請求項16】 サーバ内のアプレット記憶手段に対して最新バージョンのアプレットを記憶するアプレット記憶処理と、

該アプレットの版数に対応した版数情報を、サーバ内の版数記憶手段に記憶する版数記憶処理と、

50 クライアント内の版数情報記憶手段に対し、前記版数記

憶手段から読み出した版数情報を記憶する版数情報記憶処理と、
 所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較処理と、
 該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新処理と、
 を行うことを特徴とするアプレット最新バージョン自動設定プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、Javaアプレット等のアプレットの最新バージョンをクライアントの要求に従って自動設定するアプレットの最新バージョン自動設定システム、及び自動設定方法に関する。

【0002】

【従来の技術】今日、他のソフト上で駆動するアプレットプログラムが使用されており、特にJavaアプレットはJava言語で作成され、HTMLファイルに組み込まれてWebサーバ上で実行される。

【0003】従来、上記Javaアプレットはクライアント側において、Webページを起動する際、サーバ側よりダウンロードされ、WebブラウザのJava仮想マシン(Java virtual machine)上にキャッシュされ、使用されている。また、Javaアプレットは、通常クライアント側のメインメモリにダウンロードされ、使用されるが、一旦クライアント側のディスク(ハードディスク等)に記憶後、メインメモリに送られ実行される場合もある。

【0004】

【発明が解決しようとする課題】しかしながら、従来のシステムにおいては、アプレットをクライアント側に送信した後、サーバ側のアプレットが最新バージョンに書き換えられる場合がある。このような場合でも、一般にアプレットの更新はクライアント側に通知されない。このため、クライアント側は古いバージョンのアプレットを使用して処理を行うことになる。

【0005】このような場合、既にキャッシュされて駆動しているアプレットの更新をブラウザ側でボタン操作し、アプレットの再読み込みを行うこともできる。しかし、サーバ側でいつアプレットのバージョン更新があるか分からず、非効率な操作が必要となる。また、ブラウザの設定によっては、アプレットがキャッシュされた状態で再読み込みを実行しても、例えばディスク内に書き込まれたアプレットが読み出されるだけであり、アプレットのバージョンアップが行われない場合もある。

【0006】そこで、本発明は上記実情に鑑み、アプレット側のメソッドにバージョンアップ用のプログラムを格納し、アプレット自身の処理によってアプレットのバ

ージョンアップを実現し、自動的に最新バージョンのアプレットを使用することが可能なアプレット最新バージョン自動設定システム、及び自動設定方法を提供するものである。

【0007】

【課題を解決するための手段】上記課題は請求項1記載の発明によれば、最新バージョンのアプレットを記憶するアプレット記憶手段と、該アプレット記憶手段に記憶したアプレットの版数を記憶する版数記憶手段とを有するサーバと、該サーバから読み出したアプレットに従って処理を行い、該アプレットの版数情報を記憶する版数情報記憶手段と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数情報を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較手段と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新手段とを有するクライアントとから成ることを特徴とするアプレット最新バージョン自動設定システムを提供することによって達成できる。

【0008】ここで、上記アプレット記憶手段に記憶するアプレットは、例えばJava言語によって作成したプログラムであり、順次更新され、最新バージョンのアプレットが記憶される。また、版数情報記憶手段には、上記最新バージョンのアプレットの版数情報が記憶される。

【0009】また、上記最新バージョンのアプレットが記憶されたサーバには、例えばWebサーバが設定され、クライアントに設定されたWebブラウザとの間で通信が行われる。

【0010】一方、クライアント側は、上記サーバから読み出したアプレットによって、例えばWebブラウザを使用し、動画処理や各種表示処理を行う。また、クライアント側で使用中のアプレットの版数(バージョン)が版数情報記憶手段に記憶される。また、比較手段は、所定のイベント発生時、サーバ側の版数記憶手段から最新の版数情報を読み出し、クライアント側の版数情報記憶手段に記憶される版数情報と比較する。そして、当該比較結果が不一致の時(異なるアプレットの版数であれば)、クライアント側のアプレットの更新処理を行う。

【0011】このように構成することにより、所定のタイミングでサーバ側のアプレットの確認を行うことができ、クライアント側で使用するアプレットを自動的に最新バージョンのアプレットに設定することができる。

【0012】上記課題は請求項2記載の発明によれば、アプレットを更新し、常時最新バージョンのアプレットを記憶するアプレット記憶手段と、該アプレット記憶手段に記憶したアプレットの版数を記憶する版数記憶手段とを有し、版数情報の読み出し要求に応じ、前記版数記憶手段から版数情報を読み出し、クライアントに送信す

るサーバを提供することによって達成できる。

【0013】本請求項は上記請求項1記載のシステムに対し、特にサーバ側のシステム構成を請求するものである。したがって、上記構成の中のアプレット記憶手段や、版数記憶手段の構成は上記と同じである。

【0014】上記課題は請求項3記載の発明によれば、サーバから読み出したアプレットに従って処理を行い、該アプレットの版数情報を記憶する版数情報記憶手段と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較手段と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新手段とを有するクライアントを提供することによって達成できる。

【0015】本請求項は上記請求項1記載のシステムに対し、特にクライアント側のシステム構成を請求するものである。請求項4乃至6の記載は、前記アプレットは例えばJavaアプレットである。

【0016】このように構成することにより、サーバ側にWebサーバを設定し、クライアント側にWebブラウザを設定し、Javaアプレットを使用した処理を行うことができる。

【0017】請求項7、及び8の記載は、前記所定イベント発生時は、前記アプレットがアクティブ状態になる時である。ここで、上記アプレットがアクティブになる場合とは、例えばアイコン化されていたアプレットが駆動される場合や、他のWebページを表示していたページを、アプレットの使用ページに戻す場合である。

【0018】請求項9の記載は、上記請求項1の記載において、前記サーバとクライアント間の通信は、CORBA通信で行われる構成である。請求項10の記載は、上記請求項1の記載において、前記サーバは銀行営業センターであり、前記クライアントは支店の端末装置である。

【0019】このように構成することにより、例えば支店の端末装置では営業センター側で作成したアプレットを使用して銀行業務を行い、またアプレットを更新した際には所定のタイミングで営業センターから最新バージョンのアプレットを読み出し、窓口端末側のアプレットを更新できるので、常に最新バージョンのアプレットによって業務を行うことができる。

【0020】請求項11、及び12の記載において、前記版数情報記憶手段は、例えばメインメモリ又はディスクである。ここで、メインメモリは例えばRAMであり、ディスクは例えばハードディスク等である。

【0021】上記課題は請求項13記載の発明によれば、サーバ内のアプレット記憶手段に対して最新バージョンのアプレットを記憶するアプレット記憶処理と、該アプレットの版数に対応した版数情報を、サーバ内の版

数記憶手段に記憶する版数記憶処理と、クライアント内の版数情報記憶手段に対し、前記版数記憶手段から読み出した版数情報を記憶する版数情報記憶処理と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較処理と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新処理とを行うアプレット最新バージョン自動設定方法を提供することによって達成できる。

【0022】本請求項は上記請求項1記載のシステムに対し、アプレット最新バージョンの自動設定方法を提供するものである。このように構成することによっても、所定のタイミングでサーバ側のアプレットの確認を行うことができ、クライアント側で使用するアプレットを自動的に最新バージョンのアプレットに設定することができる。

【0023】請求項14の発明は、前記アプレットは例えばJavaアプレットである。本例は、アプレット最新バージョン自動設定方法を説明する際に、設定するアプレットがJavaアプレットであるとする構成である。

【0024】上記課題は請求項15記載の発明によれば、サーバ内のアプレット記憶手段に対して最新バージョンのアプレットを記憶するアプレット記憶機能と、該アプレットの版数に対応した版数情報を、サーバ内の版数記憶手段に記憶する版数記憶機能と、クライアント内の版数情報記憶手段に対し、前記版数記憶手段から読み出した版数情報を記憶する版数情報記憶機能と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較機能と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新機能とをコンピュータに実行させるプログラムを格納する記録媒体を提供することによって達成できる。

【0025】上記課題は請求項16記載の発明によれば、サーバ内のアプレット記憶手段に対して最新バージョンのアプレットを記憶するアプレット記憶処理と、該アプレットの版数に対応した版数情報を、サーバ内の版数記憶手段に記憶する版数記憶処理と、クライアント内の版数情報記憶手段に対し、前記版数記憶手段から読み出した版数情報を記憶する版数情報記憶処理と、所定イベント発生時、前記版数記憶手段から最新バージョンのアプレットの版数を読み出し、前記版数情報記憶手段に記憶した版数情報と比較する比較処理と、該比較結果が一致しない時、前記アプレット記憶手段から最新バージョンのアプレットを読み出し、アプレットの更新を行う更新処理とを行うことを特徴とするアプレット最新バージョン自動設定プログラムを提供することによって達成

10

20

30

40

50

できる。

【0026】本請求項は上記請求項1記載のシステムに対し、アプレット最新バージョンの自動設定処理を行うプログラムを記憶する記録媒体を特定する構成であり、このように構成することによっても、所定のタイミングでサーバ側のアプレットの確認を行うことができ、クライアント側で使用するアプレットを自動的に最新バージョンのアプレットに設定することができる。

【0027】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照しながら説明する。図1は、本実施形態のアプレット最新バージョン自動設定システムのシステム構成図である。同図において、1はサーバであり、2はサーバ1に接続されるクライアントである。尚、サーバ1には複数のクライアント2が接続されるが、代表して1のクライアントとの接続構成のみを示す。ここで、サーバ1にはWebサーバ3が設定され、Webサーバ3には最新バージョンの（最新版数の）アプレットプログラムを登録するメモリ4と、対応する最新アプレットの版数情報を登録する版数ファイル5が設定されている。上記メモリ4に登録されるアプレットはJavaを駆動する為のプログラムであり、当該アプレットによって後述するJava仮想マシンを駆動し、処理を行う。

【0028】また、版数ファイル5には対応する最新バージョンの版数の情報が、例えば図2に示す形態で登録される。図2(a)は版数ファイル5のメモリ形式を示す図であり、4バイトで構成され、下位2バイトはバージョン情報であり、上位2バイトはレベル情報である。例えば、図2(b)に示す例の場合、「V20L10」であり、版数が「20」、レベルが「10」のアプレット最新バージョンを示す。したがって、この場合同図(b)に示すように、版数ファイル5のバージョンエリアには「14」が登録され、レベルエリアには「0A」が登録される。尚、上記レベルは同じバージョンの中で、例えば開発番号を示し、本発明においては、本例のように必ずしもレベルまで最新バージョンに含める必要はない。

【0029】一方、クライアント2側では、Webブラウザ6のソフトウェア、Java仮想マシン7（パーティシャルメモリ（VM））、及びアプレット8がそれぞれ設定され、アプレット8はキャッシュ領域9に登録されている。また、処理部10には動作中のアプレットの版数情報が記憶され、例えばこの版数情報記憶エリアは所定のアドレスにセットされている。

【0030】また、Webブラウザ6は、Webサーバ3から供給されるHTML（hypertext markup language）形式のデータを画像表示し、またアプレット（Javaアプレット）8は、Java仮想マシン7を使用して処理を行う。また、上記アプレットは後述するメソッドによってサーバ1から最新バージョンのアプレットを

読み出し、キャッシュ領域9に書き込まれたアプレットを更新する。また、ディスプレイ（DSP）2'は、上記アプレットの駆動に従った表示を行う。

【0031】図3は上記サーバ1とクライアント2の関係を銀行の営業店サーバと窓口端末に適用した例である。ここで、営業店サーバ11は上記サーバ1に対応し、窓口端末12はクライアント2に対応する。また、営業店サーバ11は上述のようにWebサーバ3、メモリ4、及び版数ファイル5を有し、窓口端末12との間でCORBA（Common object request broker architecture）通信を行う。このため、CORBA通信を行うための連携アプリケーション13も搭載されている。

【0032】また、窓口端末12側には前述のWebブラウザ6が搭載され、オペレータ14によって起動する構成である。また、この場合使用するアプレットは銀行業務に関するものであり、銀行窓口でオペレータが使用する際に必要な表示や機能を備える構成である。

【0033】以上の構成のアプレット最新バージョン自動設定システムにおいて、以下にその処理動作を説明する。図4は本例の処理動作を説明するフローチャートであり、アプレットのメイン処理を説明するフローチャートである。尚、以下の説明において、サーバ1である営業店サーバ11のWebサーバ3、及びクライアント2である窓口端末12のWebブラウザ6は既に起動しているものとする。

【0034】まず、初期処理を行う（ステップ（以下Sで示す）1）。この初期処理（init）は、Webブラウザ6を起動して、サーバ1からアプレットを読み出し、クライアント2（窓口端末12）内のキャッシュ領域9に最新バージョンのアプレットをロードする処理である。そして、具体的には図5に示すイニシャルメソッドを実行する。すなわち、先ず定数変数初期化処理を行い（S2）、上記キャッシュ領域9にロードされたアプレットの版数（自版数）を設定する。この場合、例えば現在のアプレットのバージョンが、図2に示した「V20L10」であれば、「140A」のデータが書き込まれる。その後、アプレットの初期処理を行う（S3）。

【0035】次に、版数チェック処理を実行する（S4）。尚、ここで行う版数チェック処理は、特にアプレットのスタート時に行う処理であり、アプレットのスタートメソッド処理において実行される。図6はその処理を示す図であり、上記スタートメソッド内で版数チェック処理が行われる（S4'）。したがって、特にこの処理を版数チェック処理（start）で示す。

【0036】図7は上記版数チェック処理を具体的に説明するフローチャートである。先ず、サーバ1（営業店のサーバ11）に対して版数照会を行う（S5）。そして、サーバ1のメモリに記憶されている版数情報との比較処理を行い（S6）、クライアント2に読み出したアプレットの版数がサーバ1側の版数より古い場合（自版

数<サーバ版数)、後述する終了処理を行う(S7がY(イエス))。

【0037】ここで、図8は上記サーバ版数照会処理及びサーバ版数の読み込み処理を説明するフローチャートである。尚、この処理はサーバ版数照会メソッドに従って実行される。この処理は、先ずクライアント2(窓口端末12)側において、コルバ(CORBA)通信を初期化し(S5-1)、サーバ1(営業店のサーバ11)に接続する(S5-2)。そして、サーバ1(営業店のサーバ11)内の版数ファイル5からアプレットの版数情報の読み込みを行う(S5-3)。

【0038】この場合、先ずサーバ1(営業店のサーバ11)内の版数ファイル5をオープンし(S5-4)、版数ファイルの読み込み処理を行う(S5-5)。そして、読み出した版数情報を書き込み(S5-6)、版数ファイルを閉じる(S5-7)。その後、サーバ版数をアウトパラメータに設定する(S5-8)。そして、サーバ1(営業店のサーバ11)との接続を切断し(S5-9)、コルバ(CORBA)通信後処理を行う(S5-10)。

【0039】上記のように、版数照会処理を行うが、上記版数チェック処理(start)は、アプレット初期設定後のスタートメソッド処理時に行われる処理であり、クライアント3に読み出されたアプレットの版数はサーバ1側の版数と同じであり(自版数=サーバ版数)、判断(S7)はN(ノー)となる。したがって、スタートメソッドにおける版数チェック処理においては、その後イベント待ち状態に移行する(S8)。

【0040】その後、通常取引を開始する。例えば、クライアント2(窓口端末12)側からサーバ1(営業店サーバ11)に対し、例えばオンラインを介した処理を請求する。また、業務内容によってはオフライン状態で処理が行われ、窓口端末12側だけで行われる処理もある。

【0041】この間、イベントの発生を待ち(S8、S9がN(ノー))、イベントが発生すると(S9がY(イエス))、ユーザイベントか否か判断する(S10)。ここで、ユーザイベントであれば、オンライン処理が必要であるか否か判断する(S11)。ここで、ユーザイベントとは、上述のようにオンライン処理を必要とする処理であり、サーバ1(営業店のサーバ11)と接続してサーバ1側のアプリケーションを使用する処理である。

【0042】この場合、先ず版数チェック処理(S12)を実行する。この版数チェック処理(S12)は、前述の図7に示すフローチャートに従って実行される。すなわち、サーバ1(営業店のサーバ11)への版数照会(S5)、版数データの比較処理(S6)、更には使用するアプレットの版数がサーバ1側の版数と比較して古いかな否か(自版数<サーバ版数)、を比較する(S

7)。また、版数の照会処理は上記版数照会メソッドに従って実行される(図8のS5-1~S5-10)。

【0043】上記の結果、版数ファイルに読み込んだ版数情報と現在読み込まれているアプレットの版数を比較し、同じバージョンであればユーザが指定するイベント処理を実行し(S13)、イベント待ち状態に戻る(S9)。この場合、サーバ1(営業店のサーバ11)側に登録されるアプレットの版数が更新されることがなく、クライアント2(窓口端末12)側のアプレットと同じ版数であり、現在クライアント2側で使用するアプレットも最新のアプレットであり、アプレットを更新することなく、現在のアプレットによって端末業務を行うものである。

【0044】図9は上記処理を模式的に示す図である。この場合、サーバ1(営業店のサーバ11)の版数メモリには最新バージョンのアプレットが記憶されており、アプレットの版数は更新されていないので、版数情報は以前のままである。したがって、上記版数情報の照会処理(図9に示す①)を実行し、版数の一致を判断した後、取引の開始処理を行う(図9に示す②)。

【0045】一方、上記判断(S7)において、自版数<サーバ版数の関係にある場合(S7がY(イエス))、アプレットの版数がバージョンアップしたものと判断し、終了処理を実行する(S14)。この処理はデストロイ処理(destroy)であり(S14)、キャッシュ領域9に記憶されるアプレットのプログラムを消去し、サーバ1から最新バージョンのアプレットを読み出す(S15)。

【0046】また、版数ファイル5から対応するバージョンの版数情報を読み出し、クライアント2側のメモリに書き込む。また、例えば、この場合メモリにはバージョンアップした、例えば“1500”又は“140B”のデータが書き込まれる。

【0047】尚、上述の処理において、イベント発生があってもオンライン処理ではない場合(S11がN(ノー))、例えばアプレットを使用しない他のWebページに移行した処理を行う場合、対応するユーザイベント処理を実行する(S16)。

【0048】一方、例えばクライアント3が、他のWebページを表示し、又はブラウザがアイコン化している場合、アプレットは非アクティブ状態であり、この状態からWebページに戻った場合やアイコン化していたブラウザを復元した場合、システムイベントが発生する(S10がN(ノー))。この場合、前述のスタートイベントであるか否か判断する(S17)。

【0049】ここで、スタートメソッドを起動する場合(S17がYES(イエス))、前述の版数チェック処理(start)を実行する。この処理も図7に示すフローチャートに従って実行され、版数照会処理(S5、S5-1~S5-10)、版数比較処理を行い(S6、

S7)、自版数<サーバ版数である場合、アプレットの版数がバージョンアップしたものと判断し、終了処理を実行する(S7がY(イエス)、S14)。したがって、この場合にはキャッシュ領域9に記憶されるアプレットを消去し、サーバ1から最新バージョンのアプレットを読み出し、記憶する(S15)。また、版数ファイル5から対応するバージョンの版数情報を読み出し、クライアント2側のメモリに書き込む。

【0050】図10は上記処理を模式的に示す図であり、この場合サーバ1(営業店のサーバ11)側のアプレットは更新されており、版数情報が異なる。したがって、上記版数情報の照会処理(図10に示す①)を実行し、版数の不一致を判断した後、アプレットの更新処理を行い、版数情報も書き換える(図10に示す②)。

【0051】以上のように処理することによって、クライアント2(窓口端末12)側のメモリには常に最新バージョンのアプレットが記憶され、最新バージョンのアプレットによって画像表示や動画表示を行うことができる。

【0052】尚、前述のシステムイベントが発生した際、スタートイベントでなければデストロイ(destroy)イベントであるか否かを判断し(S19)、デストロイ(destroy)イベントであれば終了処理を実行する(S20)。この終了処理は前述のように、メインメモリに記憶されるアプレットのプログラムを消去し、サーバ1から最新バージョンのアプレットを読み出し、記憶する処理である。

【0053】また、上記判断(S19)において、デストロイ(destroy)イベントでなければ、指示されたイベント処理を実行する(S21)。以上のように、本発明によればアプレットをアクティブ状態にする際、又はオンライン処理を必要とするユーザイベントが発生した際等の所定の条件においてアプレットの版数チェックが行われ、最新バージョンのアプレットではない場合、必ず最新バージョンのアプレットに更新される。したがって、常に最新バージョンのアプレットがWebブラウザ6上で駆動することができる。

【0054】尚、本例においては、版数チェック処理を上記スタートメソッド等において実行する構成としたが、本例の説明に限定されるわけではない。また、本例で使用する版数のバージョン情報は、前述の図2に示すようにバージョン情報とレベル情報で構成したが、バージョン情報のみで構成してもよい。

【0055】また、本実施形態においては、本システムのサーバ1を銀行の営業店のサーバ11とし、クライアント2を銀行の窓口端末12として構成したが、本発明は上記銀行システムに限定されることなく、保険や証券等の他の分野に適用することもできる。

【0056】さらに、本発明のアプレット最新バージョン自動設定システムは、図11に示すシステムを使用し

て実現してもよい。すなわち、同図に示すようにCPU(コンピュータ)15は、内部のRAM、ハードディスク16から供給されるプログラム(データ)によって上述のシステムの実行を行う。

【0057】また、同図に示すように、内部のRAM、ハードディスク16から供給されるプログラム(データ)のみならず、外部に接続された記憶媒体との間でデータの授受を行い、例えば磁気ディスク、磁気テープ、フロッピー(登録商標)ディスク、光ディスク等の記憶媒体17から供給されるプログラム(データ)に従って処理を行う構成としてもよい。

【0058】さらに、同図に示すように、通信回線を介した提供者18から送られるプログラム(データ)を使用する構成としてもよい。

【0059】

【発明の効果】以上、詳細に説明したように本発明によれば、アプレットを使用する際には常に最新バージョンのアプレットを使用してWebブラウザを駆動することができる。

【0060】また、銀行のオンラインシステムのような、クライアントとサーバ間で連携を行い、かつ信頼性が重要な業務取引をJavaアプレットなどの可搬性のあるプログラムで実現した場合、取引の信頼性向上に大きな効果がある。

【図面の簡単な説明】

【図1】本実施形態のアプレット最新バージョン自動設定装置のシステム構成図である。

【図2】(a)は最新バージョンの版数情報を登録するメモリエリアの構成を説明する図である。(b)は(a)に示すメモリエリアに実際のデータを書き込んだ例を説明する図である。

【図3】本実施形態のシステムを営業店サーバと窓口端末に適用した例を説明するシステム構成図である。

【図4】本実施形態の処理動作を説明するフローチャートである。

【図5】イニシャルメソッドの処理動作を説明するフローチャートである。

【図6】スタートメソッドの処理動作を説明するフローチャートである。

【図7】版数チェック処理を説明するフローチャートである。

【図8】サーバ版数照会メソッドを説明するフローチャートである。

【図9】右側のリンク機構を説明する図である。

【図10】リンク機構の斜視図である。

【図11】記録媒体を使用する例を説明する図である。

【符号の説明】

1 サーバ

2 クライアント

3 Webサーバ

- 4 版数ファイル
- 5 版数情報メモリ
- 6 Webブラウザ
- 7 Java仮想マシン
- 8 (動作中の) アプレット
- 9 キャッシュ領域
- 11 ディスプレイ (DSP)
- 11 営業店サーバ

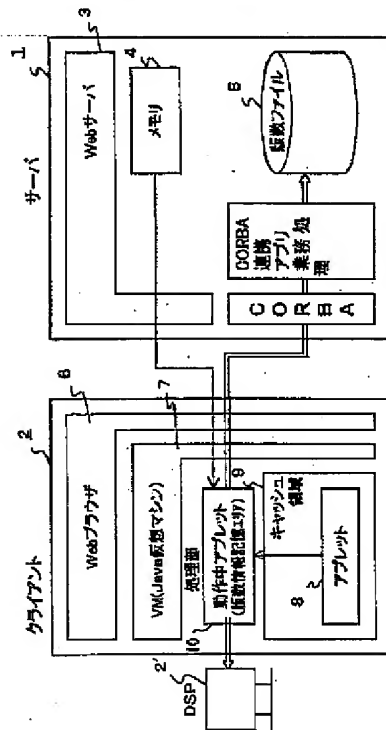
- * 12 窓口端末
- 13 連携アプリケーション
- 14 オペレータ
- 15 CPU (コンピュータ)
- 16 RAM、ハードディスク
- 17 記憶媒体
- 18 提供者

*

【図1】

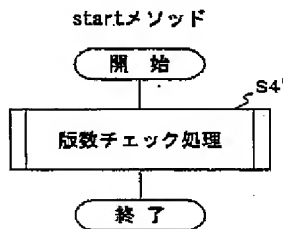
【図2】

本実施形態のアプレット最新バージョン
自動設定装置のシステム構成図

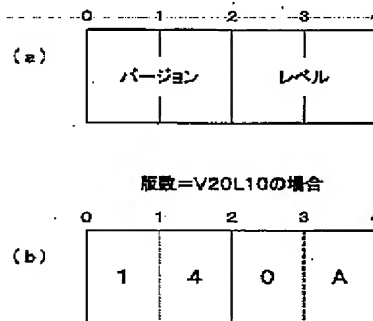


【図6】

スタートメソッドの
処理動作を説明するフローチャート



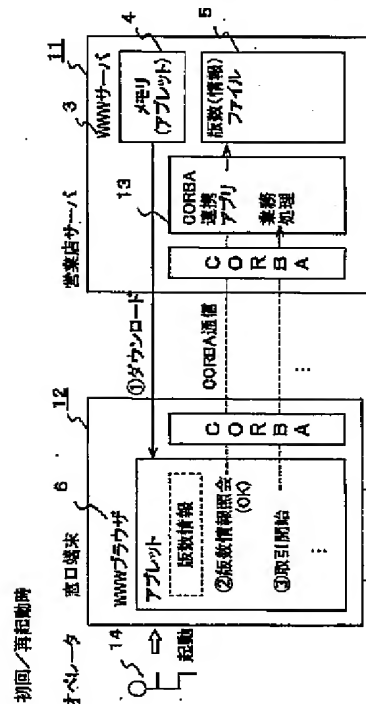
(a)は最新バージョンの版数情報を登録するメモリエリアの構成を説明する図であり、
(b)は(a)に示すメモリエリアに実際のデータを書き込んだ例を説明する図



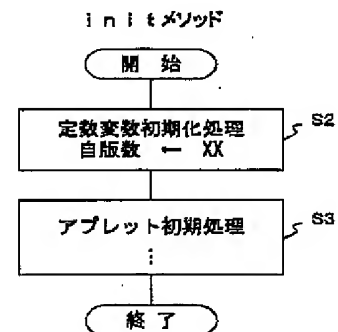
【図3】

【図5】

本実施形態のシステムを営業店サーバと
窓口端末に適用した例を説明するシステム構成図



イニシャルメソッドの
処理動作を説明するフローチャート

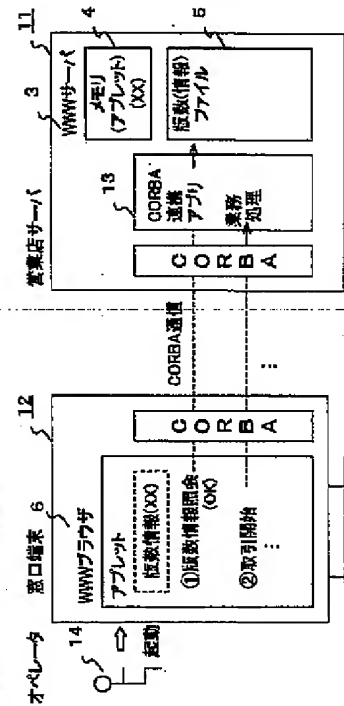
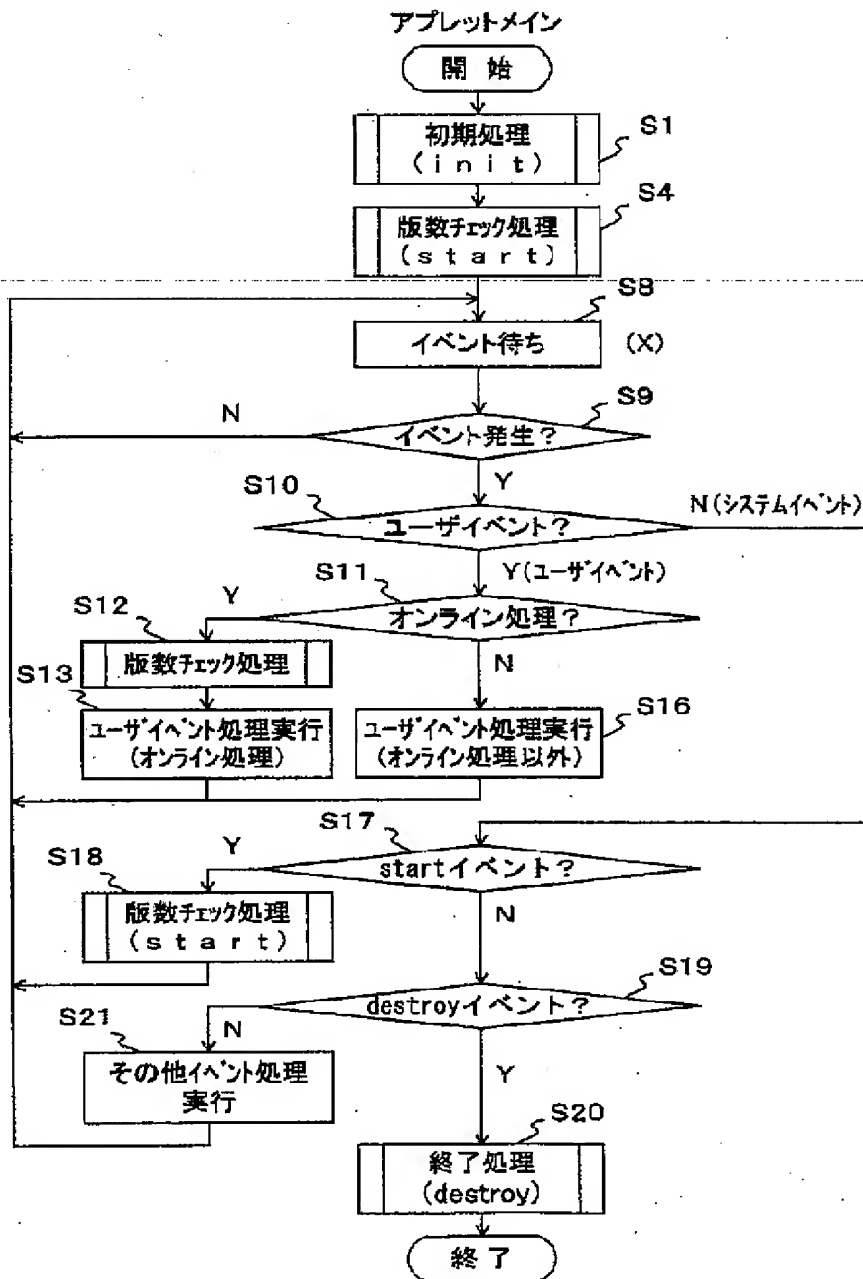


【図4】

【図9】

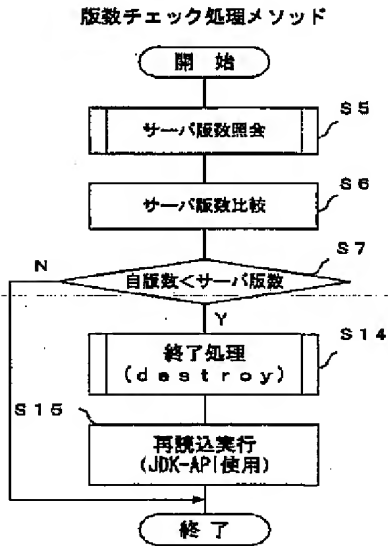
本実施形態の処理動作を説明するフローチャート

右側のリンク機構を説明する図



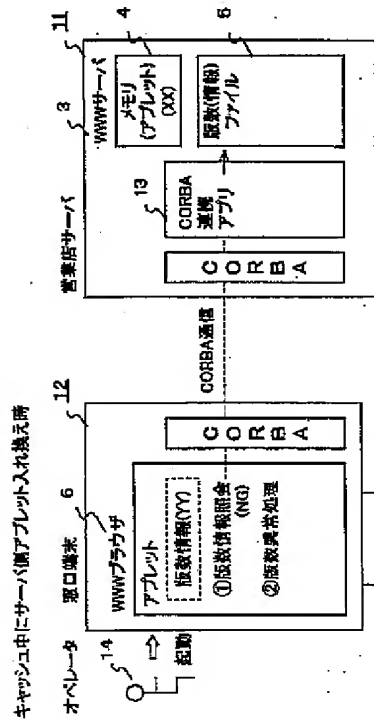
【図7】

版数チェック処理を説明するフローチャート



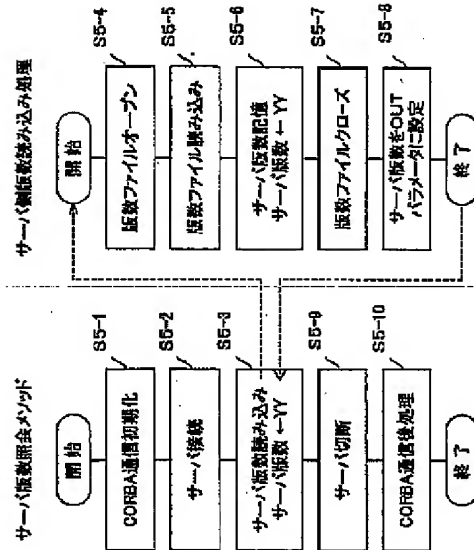
【図10】

リンク機構の斜視図



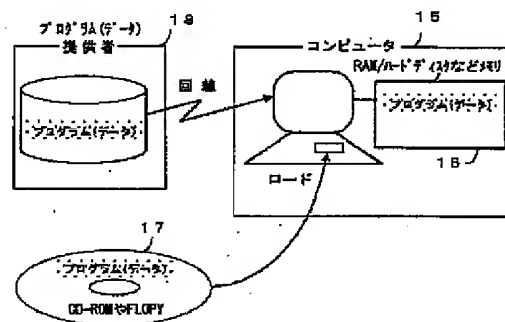
【図8】

サーバ版数照会メソッドを説明するフローチャート



【図11】

記録媒体を使用する例を説明する図



(11)

特開2001-318795

フロントページの続き

(51)Int.Cl.⁷

G06F 17/60

識別記号

240

F I

G06F 9/06

ターマコード (参考)

610Q